

Fast Dictionary Matching for Content-based Image Retrieval

Patryk Najgebauer¹, Janusz Rygał¹, Tomasz Nowak¹, Jakub Romanowski¹,
Leszek Rutkowski², Sviatoslav Voloshynovskiy³, Rafał Scherer¹

¹ Institute of Computational Intelligence, Częstochowa University of Technology
al. Armii Krajowej 36, 42-200 Częstochowa, Poland (<http://iisi.pcz.pl>)

² University of Social Sciences in Łódź
Sienkiewicza 9, 90-113 Łódź, Poland

³ University of Geneva, Computer Science Department,
7 Route de Drize, Geneva, Switzerland

{patryk.najgebauer, janusz.rygal, tomasz.nowak, jakub.romanowski,
leszek.rutkowski, rafal.scherer} @iisi.pcz.pl

Abstract. This paper describes a method for searching for common sets of descriptors between collections of images. The presented method operates on local interest keypoints, which are generated using the SURF algorithm. The use of a dictionary of descriptors allowed achieving good performance of the content-based image retrieval. The method can be used to initially determine a set of similar pairs of keypoints between images. For this purpose, we use a certain level of tolerance between values of descriptors, as values of feature descriptors are almost never equal but similar between different images. After that, the method compares the structure of rotation and location of interest points in one image with the point structure in other images. Thus, we were able to find similar areas in images and determine the level of similarity between them, even when images contain different scenes.

Keywords: content-based image retrieval, local interest points, image matching

1 Introduction

Content-based image analysis is important part of many areas of science and engineering. It can be used for face recognition [1], medical imaging [3][7][8], military science [12] and general purpose image analysis [4]. Image comparison based on their content is a complex process and still far from the excellence of the human vision. The main problem is a difference between human perception and that what can be analysed by computers. Humans focus on the remembering semantic description of an image without details as well as events, actions and objects represented by images. At the same time, they are not able to reconstruct exactly what they saw, as we remember overall image context. Humans recognize image objects by linking the situation presented on the image and

the information about it learned previously. Computers cannot find simply a relationship between images and the semantic description of objects being the content of the image. On the other hand, they are very detailed in the analysis and thanks to their precision they are able to do things such as fingerprint[5] or signature [13] recognition much faster and better than humans.

Algorithms generating local keypoints are often used for general purposes. Unfortunately they generate huge amount of information, which has to be compared afterwards. This feature implicates large complexity of computation both while generating and comparing keypoints. Using keypoints, we can focus on certain areas of the image and skip the rest.

Keypoints have also spatial relationships, which are useful in comparison of similar areas in images regarding to their rotation and shift in different images. We cannot simply search for each keypoint in an image for its equivalent in the other image as some keypoints could exist only in one image.

1.1 SURF Algorithm

One of the fastest algorithms nowadays for local interest point detection and description is the SURF algorithm [2][10]. One of the advantages of SURF over other algorithms is good performance, which is achieved by comparing areas in images and not single pixels. This approach has also a disadvantage: it is impossible to divide an image into circular areas. The algorithm can function only with square areas, which can cause some inaccuracies in estimated values, when an image is for example rotated 45 degrees. The algorithm searches for areas, in which local values of second derivatives are the highest, i.e. for local extremes. It is a typical function of algorithms of the blob detection family (Fig. 1)[11]. It also estimates the size and the rotation of the keypoint, which is important for the exploration of dependencies between keypoints. The most im-

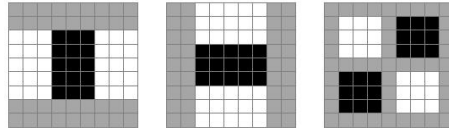


Fig. 1. SURF smallest 9x9 pixels blob detector

portant information, which is generated for each keypoint, is its descriptor. The descriptor allows to recognize a keypoint and represents local gradient around the keypoint. In the basic version of the SURF algorithm, the descriptor is built from 64 floating-point values. These values are grouped in 4 element chunks, which describe each of sixteen subregion of the descriptor: X-axis derivatives, Y-axis derivatives, modulus of the X-axis derivatives and modulus of the Y-axis derivatives

$$V = \left(\sum dx, \sum dy, \sum |dx|, \sum |dy| \right) . \quad (1)$$

Subregions create a 4x4 matrix that cover the keypoint localization in image (Fig. 2).

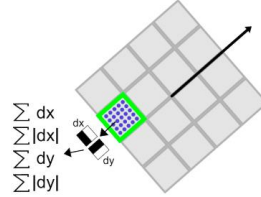


Fig. 2. Descriptor structure, 4x4 matrix of 4 values from subregion (V_{sub})

2 Description of the Problem

As already mentioned, comparison of descriptors has to be done by thresholding their difference, otherwise it will be almost impossible to match them to keypoints from a different image. For example, Table 1 presents distribution of differences in values of similar descriptors (Fig. 3) with the sum of absolute differences (SAD, L^1 norm) equal to 0.4753. In this case, we consider the keypoints with SAD lower than 0.5 as similar.

Table 1. Differences between V_{sub} of two similar keypoints descriptor

$V_{sub}x/y$	1	2	3	4
1	0.0000	0.0059	0.0031	-0.0047
2	-0.0098	0.0144	0.0349	0.0159
3	-0.0495	-0.0214	-0.0159	0.0079
4	-0.0770	-0.0062	-0.0120	-0.0173

Presented keypoints and their orientations are identical for humans, but according to the values of descriptors, they are different. After generation of keypoints in the process of comparison of two images, we have two sets of keypoints. The number of keypoints depends on the size of images and the amount of details. Often, for images larger than 1280x800 pixels, the number of keypoints exceeds 1000. The easiest and the most common approach of comparison of keypoints between images is to compare each keypoint with the rest, but when we deal with a large number of keypoints, the number of needed computations is very high. For example, 1000 of key points implicates 1 million of comparisons. To reduce the number of comparisons, keypoints should be ordered somehow and some of them should be passed over during the comparison process.

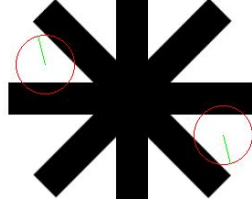


Fig. 3. An example of similar SURF keypoints with 0.47 value of difference between descriptor components

Another challenge by the estimation of similar parts is the problem of keypoints being lost during image transformations. The cause for this problem is different configuration of the same keypoints after the transformation. Usually images representing the same content contain only part of similar keypoints, another image can contain of course a different set of keypoints.

3 Method Description

For better performance the proposed method uses a special, dictionary-based form of keypoint representation [6][9]. Dictionary-based structure accelerates the comparison process by allowing to skip most of keypoint combinations.

3.1 Dictionary Creation

Before performing matching images, the method prepares images by keypoint detection and generating the dictionary structure for each single image (see Fig. 4).

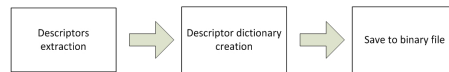


Fig. 4. Flowchart presenting the process of dictionary creation

Descriptor dictionary is created from 64 element vectors which are local interest point descriptors of an image. The method puts separate elements of the descriptor in the dictionary beginning from the first element. The dictionary is built in a similar way to the B-tree, where the first element of dictionary contains the list of first elements of descriptors.

The elements of descriptors which are similar and their values do not exceed estimated limits, are grouped and will be represented as a single element of the dictionary. An example of grouping is presented in Fig. 5 for the first element of descriptors with the number between 2 and 6. The rest of descriptor elements,

from which another elements are built, are derivatives of the first group. Thanks to grouping, we can decrease the number of similar, duplicated elements of descriptors. Thanks to the presented approach, building index of descriptors is also

		Number of descriptor element				
		1	2	3	4	5
Descriptor Number	1	-0.1021	0.2031	0.1229	-0.0551	0.2168
	2	0.0451	0.1846	0.2229	-0.0551	-0.0720
	3				0.1779	-0.0068
	4					0.2168
	5				-0.1021	-0.0067
	6	0.2687	0.4722	-0.0551	0.2168	0.1780
	7		0.1846	-0.0067	0.0205	-0.0068
	8		0.5151	0.2168	0.1779	0.2168
	9		0.1847	-0.0068	-0.0551	-0.0721
	10	0.4552		0.0205	-0.0068	0.2168

Fig. 5. Exemplary part of the descriptor dictionary

faster, especially when we deal with a very large number of descriptors. The rest of data of keypoints such as position, size or orientation are contained in the last part of the word associated with the descriptor. The last step of the process of creation of the dictionary is conversion of data to a binary file as it is sufficient to generate the dictionary only once.

3.2 Comparison Between Descriptors and Dictionary

Every image from the analyzed set has its own descriptor dictionary stored in a form of a binary file (see Section 3.1). Now, let us assume that we have a new query image and we want to find similar images in the large collection of images. The first step is to create a dictionary of its feature descriptors and store it in a binary file. Fig. 6 presents a flowchart of such image retrieval.

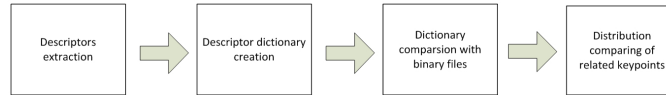


Fig. 6. Flowchart of image retrieval searching in the set of images

The next step is a comparison of the query image dictionary with the dictionaries from the binary files. Descriptors values are similar if their sum of absolute differences (SAD) is less than the threshold. Comparison of two dictionaries is presented in the Fig. 7, where the dark background represents a common part.

		Number of descriptor element				
		1	2	3	4	5
Descriptor Number	1	-0.1021	0.2031	0.1229	-0.0551	0.2168
	2	0.0451	0.1846	0.2229	-0.0551	-0.072
	3				0.1779	-0.0068
	4					0.2168
	5					0.0205
	6					0.1780
	7	0.2687	0.5151	0.2168	0.1779	0.2168
	8		0.1846	-0.0067	0.0205	-0.0068
	9	0.4552	0.1846	-0.0068	-0.0551	-0.0720
	10			0.0205	-0.0068	0.2168
Dictionary form binary file						
		Number of descriptor element				
		1	2	3	4	5
	1	0.042	0.5111	-0.1021	-0.0067	0.0205
	2			-0.0551	0.2168	0.1780
	3	0.1187	0.1846	-0.0067	0.0205	-0.0068
	4			0.5151	0.2168	0.1779
	5	0.2687	0.1846	-0.0205	-0.0068	0.2168
	6			-0.0068	0.03	-0.0068

Fig. 7. Exemplary part of two compared dictionaries

3.3 Matching Determined Sets of Keypoints

The dictionary comparing process returns a set of pairs of similar keypoints. The next step is to examine keypoint distribution between images. Each pair will be excluded, if their distribution in the relation to the rest of pairs indicates wrong connection. Fig. 8 describes an example of keypoint distribution between two images. Each point has its own counterpart in the second set. The method

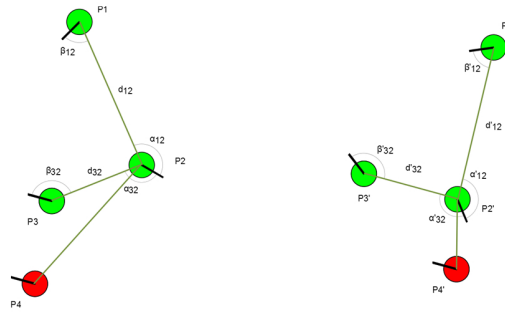


Fig. 8. Example of keypoints pair checking by mapping them between images.

compares the direction and the distance between keypoints from the same set. For example, angles β_{12} and α_{12} have the same value as β'_{12} and α'_{12} from the

second set. Distances d_{12} and d_{12}' are also similar. Thus, in this case we can assume, that points P1 and P2 are related. In other case we mark points as not related, e.g. P4 and P4'.

4 Experimental Results

In this section we show some examples of the proposed method for content-based image retrieval on the test images presented in Fig. 9. For better presentation, we chose images, which are only slightly different.



Fig. 9. Images used in the experiments

Table 2 contains test results of comparison between each image with all other from Fig. 6. "No. of points" column is the number of descriptors extracted from the image. "Matched" column is the number of related descriptors between current and all other images. "Comparisons" is the number of descriptors compared by using the dictionary. "Combination" is the number of all possible combinations of descriptors between images. As we can see, the number of comparisons in the proposed method is significantly smaller in relation to the number of all combinations. In our tests, the number of compared descriptors is only 0.18% of all possible combination.

Table 2. Results of comparisons between dictionaries.

Images	No. of points	Matched	Comparisons	Combinations	Performance
01	367	225	19522	3733124	0.52%
02	257	26	3691	2614204	0.14%
03	727	103	15373	7395044	0.21%
04	80	101	1747	813760	0.21%
05	408	112	10773	4150176	0.26%
06	24	22	413	244128	0.17%
07	729	0	0	7415388	0.00%
08	414	20	7676	4211208	0.18%
09	845	20	7674	8595340	0.09%
10	359	128	5137	3651748	0.14%
11	318	128	5107	3234696	0.16%
12	213	44	3815	2166636	0.18%
13	891	52	13049	9063252	0.14%
14	785	61	19567	7985020	0.25%
15	435	162	10068	4424820	0.23%
16	295	95	10575	3000740	0.35%
17	489	154	10408	4974108	0.21%
18	650	116	14754	6611800	0.22%
19	417	186	13569	4241724	0.32%
20	464	104	13479	4719808	0.29%
21	1005	5	134	10222860	0.00%

Fig. 10 presents the results of search for common set of keypoints from image number 17. The largest image is our query image. The others are found similar images. Related points are marked on each image. Larger points are the centers of keypoints that describe common area.

Table 3 presents detailed values from comparison procedure between images from Fig. 10. Only part of keypoints were connected, but this number allows selecting a common part of both images. In this case, a single image has been incorrectly marked as related to the query image. It was caused by a similarity between descriptors and their orientation.



Fig. 10. Sample of detected groups of descriptors between images

Table 3. Results of comparison between images from Fig.10

No. image (query)	No. of keypoints	No. image (compared)	No. of keypoints	Math Comparisons	
17	489	4	80	31	290
17	489	15	435	17	1491
17	489	18	650	20	2197
17	489	19	417	57	1708
17	489	20	464	23	1723

5 Conclusion

Analysing results of our test, we can say that creation of the dictionary allows to significantly decrease the number of operations, which have to be done in the process of image comparison. In our case, the number of operation has been reduced to the 0.18 of all operations. The approach obtains better results in the case of larger sets of images. Images related to the query image can be found much faster in comparison to the standard all-to-all approach. Moreover, saving the dictionary in a binary file allows for more efficient image multiple comparison and reuse of data.

Acknowledgements

This work was supported by the Polish National Science Centre (NCN) within project number DEC-2011/01/D/ST6/06957.

References

1. Akhtar, Z., Rattani, A., Foresti, G.L.: Temporal analysis of adaptive face recognition. *Journal of Artificial Intelligence and Soft Computing Research* **4**(4) (2014) 243–255
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). *Comput. Vis. Image Underst.* **110**(3) (June 2008) 346–359

3. Bruzdziński, T., Krzyżak, A., Fevens, T., ukasz Jelen: Webbased framework for breast cancer classification. *Journal of Artificial Intelligence and Soft Computing Research* **4**(2) (2014) 149–162
4. Chu, J.L., Krzyżak, A.: The recognition of partially occluded objects with support vector machines and convolutional neural networks and deep belief networks. *Journal of Artificial Intelligence and Soft Computing Research* **4**(1) (2014) 5–19
5. Drozda, P., Sopyła, K., Górecki, P.: Online crowdsourcing system supporting ground truth datasets creation. In: *Artificial Intelligence and Soft Computing - 12th International Conference, ICAISC 2013, Zakopane, Poland, June 9-13, 2013, Proceedings, Part I*. (2013) 532–539
6. Edelkamp, S., Schroedl, S.: *Heuristic Search: Theory and Applications*. Elsevier Science (2011)
7. Karimi, B., Krzyżak, A.: A novel approach for automatic detection and classification of suspicious lesions in breast ultrasound images. *Journal of Artificial Intelligence and Soft Computing Research* **3**(4) (2013) 265–276
8. Makinana, S., Malumedzha, T., Nelwamondo, F.V.: Quality parameter assessment on iris images. *Journal of Artificial Intelligence and Soft Computing Research* **4**(1) (2014) 21–30
9. Najgebauer, P., Nowak, T., Romanowski, J., Gabryel, M., Korytkowski, M., Scherer, R.: Content-based image retrieval by dictionary of local feature descriptors. In: *2014 International Joint Conference on Neural Networks, IJCNN 2014, Beijing, China, July 6-11, 2014*. (2014) 512–517
10. Pena, M.: *A Comparative Study of Three Image Matching Algorithms: Sift, Surf, and Fast*. BiblioBazaar (2012)
11. Wang, L., Ju, H.: A robust blob detection and delineation method. In: *Education Technology and Training, 2008. and 2008 International Workshop on Geoscience and Remote Sensing. ETT and GRS 2008. International Workshop on*. Volume 1. (Dec 2008) 827–830
12. Wang, X., Japkowicz, N., Matwin, S.: Automated approach to classification of mine-like objects using multiple-aspect sonar images. *Journal of Artificial Intelligence and Soft Computing Research* **4**(2) (2014) 133–148
13. Zalasinski, M., Cpalka, K.: New approach for the on-line signature verification based on method of horizontal partitioning. In: *Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L., Zurada, J., eds.: Artificial Intelligence and Soft Computing. Volume 7895 of Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2013) 342–350